

# RECOPILACIÓN ENUNCIADOS DE TADs DE EXÁMENES

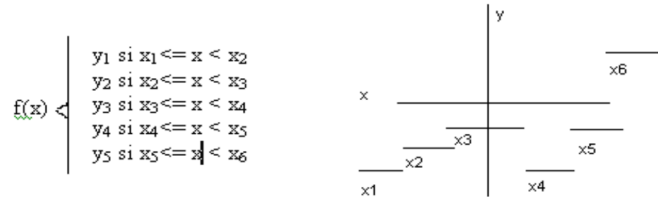
**Análisis de Algoritmos y Estructuras de Datos  
2º Curso - Primer Semestre**

**Grado en Ingeniería Informática  
Universidad de Cádiz**

# TAD ESCALONADA

## PROBLEMAS

1. Una función  $f: \mathbb{R} \rightarrow \mathbb{R}$  es escalonada cuando consiste en una sucesión de funciones constantes definidas en subintervalos disjuntos y contiguos, es decir,  $f$  puede definirse mediante condiciones de la forma  $f(x) = y_i$  si  $x_{i-1} \leq x < x_i$ , donde los valores  $y_i$  son distintos para subintervalos adyacentes. Cada uno de los puntos  $(x_i, y_i)$  en los que la función cambia de valor se llama salto o escalón.



(40%) Especificar un TAD para las funciones escalonadas con un número finito de saltos, que incluya las siguientes operaciones:

- Crear una función constante  $f(x) = y_i$  definida a partir de  $x_i$
- Añadir un nuevo salto en el punto  $(x, y)$  a una función. Si ya existe un salto en las coordenadas  $x$ , se sustituye por el nuevo.
- Eliminar el escalón definido en el subintervalo al que pertenece la coordenada  $x$ . El escalón anterior al eliminado se prolonga hasta el siguiente.
- Calcular el valor de una función en un punto dado
- Calcular el valor mínimo de una función escalonada
- Calcular el valor máximo de una función escalonada
- Hacer una translación de una función  $w$  unidades horizontales y  $z$  unidades verticales, siendo  $w$  y  $z$  números reales.
- Destruir una función

# TAD COCINA

Una empresa de muebles de cocina necesita un TAD para representar el conjunto de muebles colocados en la pared de una cocina. Una cocina se crea con una longitud positiva, y un mueble colocado en la pared se identifica con el par formado por su posición (distancia desde su lateral izquierdo al extremo izquierdo de la pared) y su anchura (la profundidad y altura no tienen interés, pues son iguales para todos los muebles).

El TAD debe soportar las siguientes operaciones:

- Crear una cocina vacía con una longitud dada.
- Determinar si un mueble de una cierta anchura puede colocarse en una posición dada.
- Añadir un mueble de una determinada longitud a una posición dada.
- Devolver el mueble  $i$ -ésimo de la cocina empezando a contar por la izquierda.
- Eliminar el mueble  $i$ -ésimo de la cocina, si existe.
- Mover el mueble  $i$ -ésimo de la cocina (si existe) hacia la izquierda, hasta que se junte con el mueble  $(i - 1)$ ésimo o el extremo izquierdo de la pared.
- Destruir la cocina.

Se pide:

- a) Especificar el TAD Cocina (30 %).
- b) Diseñar una estructura de datos adecuada para representar el TAD Cocina (10 %).
- c) Implementar las operaciones anteriormente descritas (60 %).

## TAD TEXTO

(Definir prototipos, operaciones del TAD. No hay carácter de salto de línea, el texto te lo dan a partir de la estructura)

- a) Diseña una estructura de datos para almacenar en memoria un texto del cual desconocemos el número de líneas y de caracteres por línea.  
Sabemos que el texto solo consta de caracteres alfanuméricos y signos de puntuación, es decir, en él no aparecen caracteres de control como fin de línea, retorno de carro... Sin embargo, hay caracteres especiales que pueden aparecer un número indeterminado de veces.  
@ indica el borrado del carácter anterior. En caso de que aparezcan varios consecutivos, solo tiene efecto el primero; el resto aunque se almacene, se ignora. Este carácter no aparecerá al principio de ninguna línea.  
# indica el borrado de la línea completa. Si aparecen varias igual, solo aparecerá al final de la línea.
- b) Escribe una función MostrarTexto(). Al realizar este volcado a pantalla no te preocupes por el número de caracteres ni de líneas.

Ejemplo: <u>Texto original</u>	<u>Texto impreso</u>
El munucipl@io	El municipio
ligare@s@	ligar

## TAD HOSPITAL

Especificad e implementad el *TAD Hospital*. Este TAD representa los pacientes (código de paciente) ingresados en el mismo. La gravedad de los pacientes se indica mediante un dígito decimal (en orden decreciente de gravedad). Si un paciente tiene una gravedad entre 1 y 5, estará ingresado en U.C.I. (Unidad de Cuidados Intensivos), si su gravedad se encuentra entre 6 y 9, estará en planta. Sobre decir que la gravedad 0 implica que el paciente ingresa en la morgue. Obviamente, la capacidad de la U.C.I. y de la planta es limitada, por lo tanto, si un paciente tiene que ingresar en U.C.I. y ésta se encuentra completamente ocupada, el paciente menos grave de la misma pasará a planta, y si sucede lo mismo con un ingreso en planta, se dará de alta al paciente con mejor estado de salud. Esta situación anómala (tener pacientes con gravedad U.C.I. en planta, será resuelta lo antes posible, volviendo dichos pacientes a U.C.I. a la primera oportunidad).

Las operaciones del TAD serán las siguientes:

- CrearHospital.
- Ingreso.
- Alta.
- Muerte.
- Pacientes en U.C.I.
- Pacientes en Planta.
- Pacientes de una gravedad determinada.
- DestruirHospital.

## TAD TREN

Realizar la especificación e implementación de un TAD Tren que sirva para representar secuencias (de longitud arbitraria) de vagones (se supone que el tipo vagón está predefinido). Si la secuencia no está vacía, hay un vagón del tren denominado "activo" sobre el que puede actuar un robot manipulador. Las operaciones que se piden son:

- Crear un tren.
- Desplazar a la izquierda: Mueve un tren una posición a la izquierda de manera que el nuevo vagón activo es el que está inmediatamente a la derecha del actual. Si no hay vagón a la derecha no se hace nada.
- Desplazar a la derecha: Simétrica a la anterior.
- Eliminar el vagón activo: Se suprime el vagón activo y se convierte en activo el que está inmediatamente a su derecha. Si éste no existe, se activa el de su izquierda. Y si éste último tampoco existe, el tren queda vacío.
- Insertar vagón activo: Añade un nuevo vagón al tren desplazando el actual vagón activo y los siguientes una posición hacia la derecha.
- Observar el vagón activo: Una cámara fija situada sobre el robot devuelve el vagón activo.
- Tren vacío: indica si el tren tiene vagones.

**Nota: Se prohíbe expresamente utilizar una estructura doblemente enlazada. Y se exige que el coste de todas las operaciones sea constante, es decir, independiente del número de vagones del tren.**

Se valorará la claridad de la especificación y la eficiencia de las operaciones.

## TAD DICCIONARIO

2º Crear diccionario inglés-español, en que palabra inglesa puede tener varias traducciones:

a) Especifica TAD diccionario:

- Crear uno vacío
- Insertar traducción español de una palabra inglesa
- Eliminar Traducción español->inglés
- Consultar TODAS las traducciones al español de una inglesa
- Destruir diccionario

b) Implementar

**\*\*Se puede usar la biblioteca string.h\*\***



## TAD SÍMBOLO

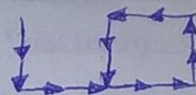
### Problema

• Una manera de representar (o, en símbolos) por línea continua y escritas mediante secuencia de trazos, D(derecha), I(Izquierda), S(subir), B(bajar)

ejemplo:



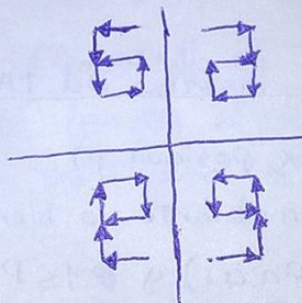
simbolo: IBBDSI



Simbolo: BBDDDDSSIIIBB

a) Escribir la especificación del TAD simbolo con las siguientes operaciones:

- Crear Simbolo vacío
- Añadir trazo al final de un simbolo.
- Deshacer los n últimos trazos de un simbolo
- Tres operaciones que devuelvan el simbolo simétrico respecto (x), respecto (y) y ambas ejes (x e y)



b) Realiza la implementación del TAD simbolo, obligando utilizar algunas de las TAD vistas en clase, justificando la elección del mismo así como su implementación más adecuada.

## TAD CONSULTORIO

La dirección de un hospital quiere informatizar su consultorio médico con un programa que realice las siguientes operaciones:

- Generar consultorio vacío.
- Dar de alta a un nuevo médico.
- Dar de baja a un médico.
- Poner a un paciente en la lista de espera de un médico.
- Consultar al paciente a quien le toca el turno para ser atendido.
- Atender al paciente que le toque por parte de un médico.
- Comprobar si un médico determinado tiene o no pacientes en espera.

Diseñar la ED para representar el TAD e implementar las operaciones anteriores.

**\*NO OLVIDAR REPASAR OTROS TADs Y EJERCICIOS DE LAS PRÁCTICAS.**